REMARKS

Drawings

The drawings are objected to as failing to comply with 37 CPR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: 44 of fig 1, 80, 82, 84, 86, 88, 90, 92, 96, 98, 100, 102 of fig 3, 450 of fig. 4, 509, 510, 511, 521,of fig. 5. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Applicants respectfully respond to the objections to the drawings by calling the examiner's attention to the fact that most, if not all, of the numbers noted by the examiner are, in fact, included in the drawings. Thus, the numerals 44, 80, 82, 84, 86, 88, 90, 92, 96, 98, 100, 102, 509, 510, 511, 521 all appear in the drawings. Applicants herewith submit Sheets 1, 3 and 5 with the numbers circled for clarification. If there is a discrepancy between the figures herein submitted and the drawing sheets as filed, applicants respectfully request that the examiner advise the undersigned. Furthermore, page 19 has been amended to delete all references to the numeral 450 appearing thereon. The correct part number is 411, which is shown in Figure 4. Applicants respectfully submit that this action is dispositive of the drawing issue.

Claim Rejections - 35 USC § 103

Claims 1 - 3; 6 -16; 18 - 20, 23 - 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung, US patent no. 5,404,469.

As to claim 9, Chung teaches a network processing system including an embedded processor complex for controlling the programmability of a network processor, the complex including a plurality of protocol processor units (PPU), each PPU containing:

at least one core language processor (processor 10, cal. 2 lines, 30 - 50), each CLP having at least two threads (threads, col. 2 lines 30 - 32),

a plurality of coprocessors (plurality of function units, cal. 2 lines 30 - 50) for executing specific tasks for the system.

Chung does not teach multiple coprocessor interfaces to access and share the resources of the coprocessors with each CLP.

It would have been obvious for one skilled in the art by the time of the invention to recognize that there must have interfaces between core processor and coprocess rs for communication.

As to claim 10, Solomon as modified by Chung teaches coprocessor interfaces are dedicated to support the code threads (threads can run between core processor and coprocessors through interface).

As to claim 11, Chung teaches checksum coprocessor (integer unit. col. 2 lines 30 - 45), datastore coprocessor (load/store unit). Stringcopy coprocessor, counter processor are generic coprocessors executing specific tasks.

As to claim 12, 13, 14, Chung teaches the network processing system of claim 10 further including a FIFO buffer (each buffer contains two horizontal instruction words and each horizontal instruction word contains a section corresponding to each function unit, cal. 4 lines 5 - 10) between each thread and at least one of the coprocessors

As to claim 15, 16, Chung teaches the network processing system including specific operating instructions (instructions shown in the instruction buffers are scheduled by the compiler statically ....selecting threads, cal. 4 lines 11 - 20) executed by the threads of the CLPs which result in commands to control coprocessor operation, which commands flow through the interface between the CLPs and the coprocessors,

As to claim 1 and 18, see claim 9 above.

As to claim 2 and 19, see claim 10 above.

As to claim 3 and 20, see claim 11 above.

As to claim 6 - 8, 23 - 25, see claim 12 - 14 above.

As to claim 26, 27, see claim 15, 16 above.

As to claim 28, Chung teaches the method according to claim 27 wherein the execution is either direct or indirect (the instructions issued to the function unit in the third clock cycle. By the end of the third clock cycle, all the instructions from the three threads are issued, col. 4 lines 10 - 60).

Applicants respectfully traverses the rejection of claims 1-3, 6-16, 18-20 and 23-28 on the basis that these claims are patentably distinguishable over the teachings of Chung et al under 35 U.S.C. 103(a).

To briefly summarize, the system of claims 1 and 9 of the present application contain the following basic elements for controlling the programmability of a network processor:

1. A plurality of protocol processor units (PPUs);
2. At least one core language processor (CLP) associated with each PPU; and
3. A plurality of coprocessors for executing specific tasks for the system.

Thus, it can be seen that applicants are claiming a "network processing system", or an "embedded processor complex", or the use of such a complex. Contrary to the contention of the examiner, Chung et al do not teach a network processing system. To the contrary, "network processing system" is mentioned nowhere in Chung et al. Chung et al teach at (column 5, lines 44-47) that a "static interleaving technique is utilized to resolve contention in a very long instruction word multi-threaded microprocessor architecture."

The examiner then suggests that Chung et al teach "at least one core language processor (column 2, lines 30-50)." This is not so. Chung et al teach a superscalar multi-threaded architecture where multiple units (threads) dispatch instruction execution to multiple execution units (FUs) (column 3, lines 36-54). Chung et al are primarily concerned with static scheduling as performed by a compiler of these FUs. (column 3, lines 8-35, column 4, lines 11-34). In order to perform such optimal static scheduling, the compiler must be aware of the execution details of each instruction. There is no mention in Chung et al of a "core" instruction set coupled with a co-processor instruction set as taught and claimed by applicants in their application.

The examiner then states that Chung et al teach a "plurality of coprocessors" (column 2, lines 30-50). Chung et al at column 2, lines 30-50, describe "function units" (FU) which are instruction level, i.e. load/store, integer unit, and logic unit as in a classic super-scaler computer architecture. These units are tightly integrated into the pipeline execution structure with a fixed execution time so as to allow an optimizing compiler to effectively schedule the pipeline. FUs are not coprocessors as the term is widely used. In contrast, the present application teaches "coprocessors useful for specific tasks", such as

tree search coprocessor, checksum coprocessor, and enqueue coprocessor, etc. These coprocessors used by applicants certainly are not generic low-level FUs as taught by Chung et al. An integer unit of the type used by Chung et al is widely accepted to mean "add/subtract/shift/logical" on fixed length integers (32-bits, 64-bits, etc.). The checksum coprocessor used in the present invention computes checksums on arbitrarily long data blocks. A load/store unit is generally accepted to mean loading and storing of a single data item of fixed size: 8, 16, 32, 64 bit etc. On the other hand, applicants' data store coprocessor loads blocks of data from packet memory into the coprocessor's local memory.

Applicants conclude that the prior art teaching of Chung et al falls far short of a teaching that would render the claims of the present invention obvious under a single reference §103 rejection. It is not enough that one may modify a reference, but rather it is required that a second reference or some objective teaching suggest such modification of the first reference.

The law is quite clear that in order for a claimed invention to be rejected on obviousness, the prior art must suggest the modifications sought to be patented; In re Gordon, 221 U.S.P.Q. 1125, 1127 (CAFC 1984); ACS Hospital System, Inc. v. Montefiore Hospital, 221 U.S.P.Q. 929, 933 (CAFC 1984). The foregoing principle of law has been followed in Aqua-Aerobic Systems, Inc. v. Richards of Rockford, Inc., 1 U.S.P.Q. 2d, 1945 (D.C. Illinois 1986). In the Aqua-Aerobic's case, the Court stated that the fact that a prior reference can be modified to show the claimed invention does not make the modification obvious unless a prior reference suggests the desirability of the modification.

In In Re Oetiker, 24 U.S.P.Q. 2nd 1443, 1445 (CAFC 1992) held:

> "There must be some reason, suggestion, or motivation found in the prior art whereby a person of ordinary skill in the field of the invention would make the combination. That knowledge can not come from the applicant's invention itself."

Thus, it is clear that where an individual reference does not teach the entire invention, then the modification which the invention represents must be suggested and motivated by some other reference or through some objective teaching and cannot come from the application itself. Hence, there clearly can be no suggestion of modifications in any way, let alone as suggested by the Examiner inasmuch as there is but one reference cited.

Applicants respectfully submit that the rejection of the three independent claims 1, 9 and 18 is unsupportable and that that all three of these claims should be allowed. Based on the belief that the rejection of the independent claims is unfounded, then all of the other claims covered by this first rejection, namely claims 2, 3, 6, 7, 9-16, 19, 20, and 23-28 are likewise patentable and the corresponding rejection should be withdrawn.

Claims 4, 5, 17, 21, 22, 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung US patent no, in view of Bitar, US patent no. 5,928,322.

As to claim 17 and 29, Chung does not teach latency events.

Bitar teaches the network processing system according to claim 15 wherein the instructions enable the system to identify long latency events (latency, cal. 4 lines 55 - 67) and short latency events (low-latency, cal. 3 lines 59 - 60) according to the expected response time to access data in response to a particular coprocessor command, and to grant full control to another thread when execution of an active thread stalls due to a long latency event, or to grant temporary control to another thread when execution of an active thread stalls due to a short latency event.

It would have been obvious to apply the teaching of Bitar to Chung's system because latency events used to control timing for executing threads.

In light of the arguments submitted above in connection with independent claims 1, 9 and 18, applicants respectfully submit that the rejection of the foregoing claims, all of which depend from these independent claims, should likewise be withdrawn.

As to claim 4, 5, 21, 22, Bitar teaches in the operation according to claim 3 further including a coprocessor execution interface arbiter to determine the priority between multiple data threads (changes to the priority of one or more real-time threads is another event, col. 4 lines 5 - 20).

In light of the arguments submitted above in connection with independent claims 1, 9 and 18, applicants respectfully submit that the rejection of the foregoing claims, all of which depend from these independent claims, should likewise be withdrawn.

CONCLUSION

Applicants respectfully submit that all necessary corrections have been made to the specification to overcome the Section 112 issues. Furthermore, they have presented arguments that are sufficient to refute the contention by the examiner that the claims of the present application do not cover patentable subject matter. The principal reference, Chung et al, is prior art, but the relevance of the patent to the claimed subject matter is remote at best. Accordingly, applicants respectfully request that the examiner reconsider his position concerning patentability and withdraw his rejections.

Respectfully submitted,

Date: Nov. 18, 2003

James A. Lucas, Reg. No. 21,081
Attorney for Applicants
DRIGGS, LUCAS, BRUBAKER & HOGG CO., L.P.A.
8522 East Avenue
Mentor, Ohio 44060
(440) 205 3600 (phone)
(440) 205 3601 (fax)
E-mail: jlm@driggslaw.com

JAL/

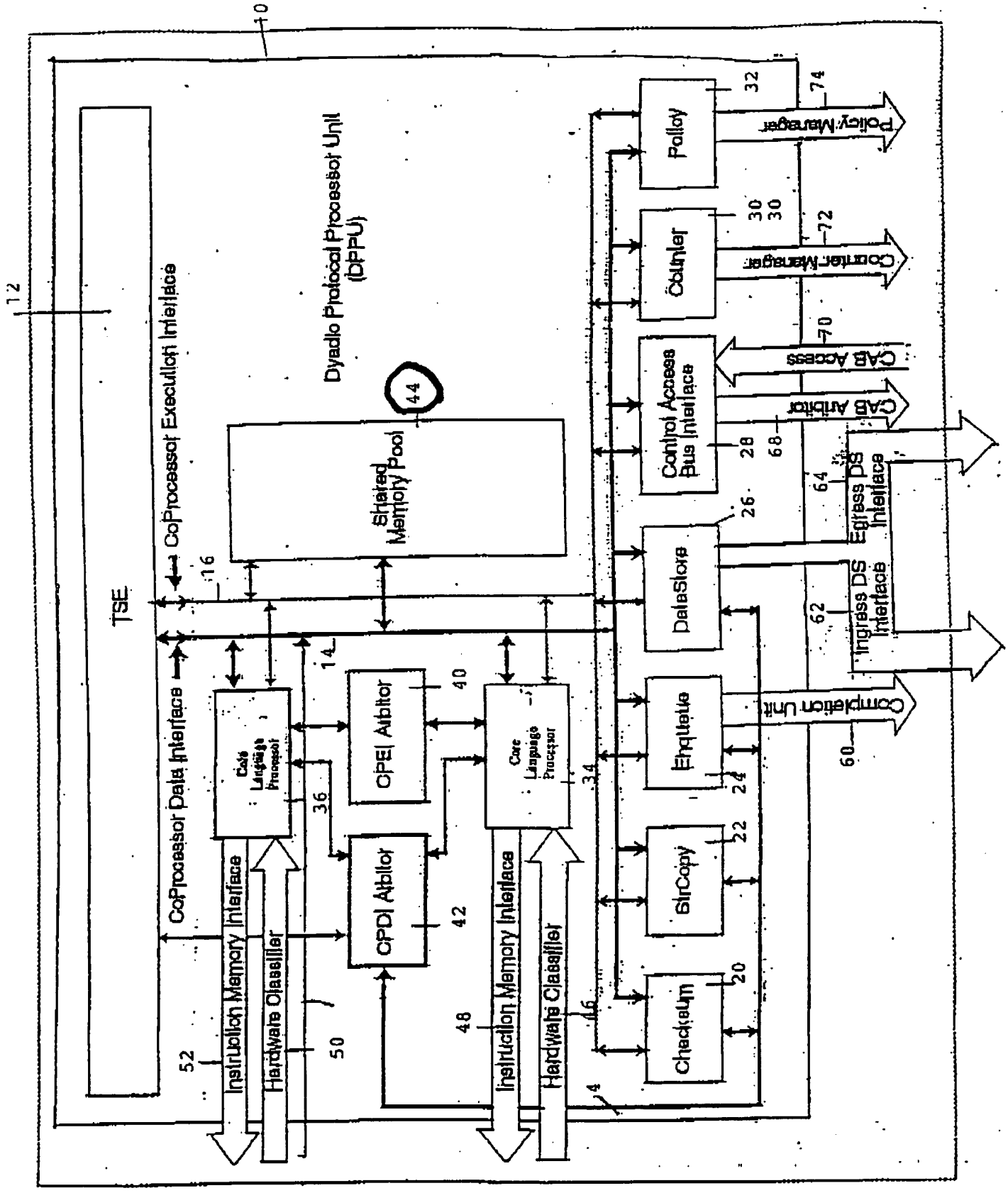

10
12
TSE
CoProcessor Execution Interface
Dyadic Protocol Processor Unit (DPPU)
44
Shared Memory Pool
16
CoProcessor Data Interface
14
Data Language Processor
36
Instruction Memory Interface
Hardware Classifier
52
50
CPEI Arbitor
40
CPDI Arbitor
42
Core Language Processor
34
48
Instruction Memory Interface
Hardware Classifier
46
4
Policy
32
74
Policy Manager
Counter
30
30
72
Counter Manager
70
CAB Access
CAB Arbitor
Control Access Bus Interface
28
68
64
Egress DS Interface
DataStore
26
62
Ingress DS Interface
Enqueue
24
Completion Unit
60
SlrCopy
22
Checksum
20

FIGURE 1

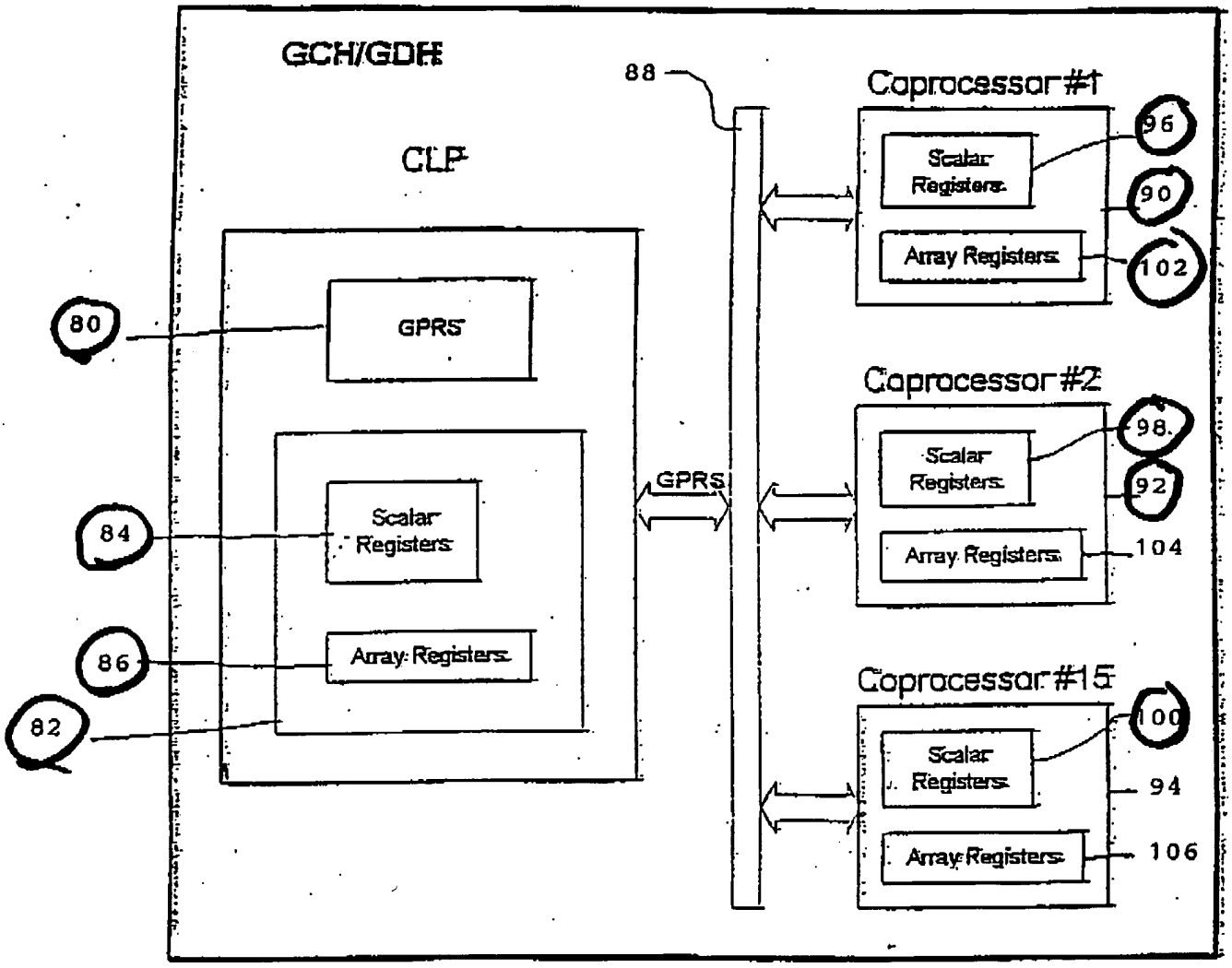
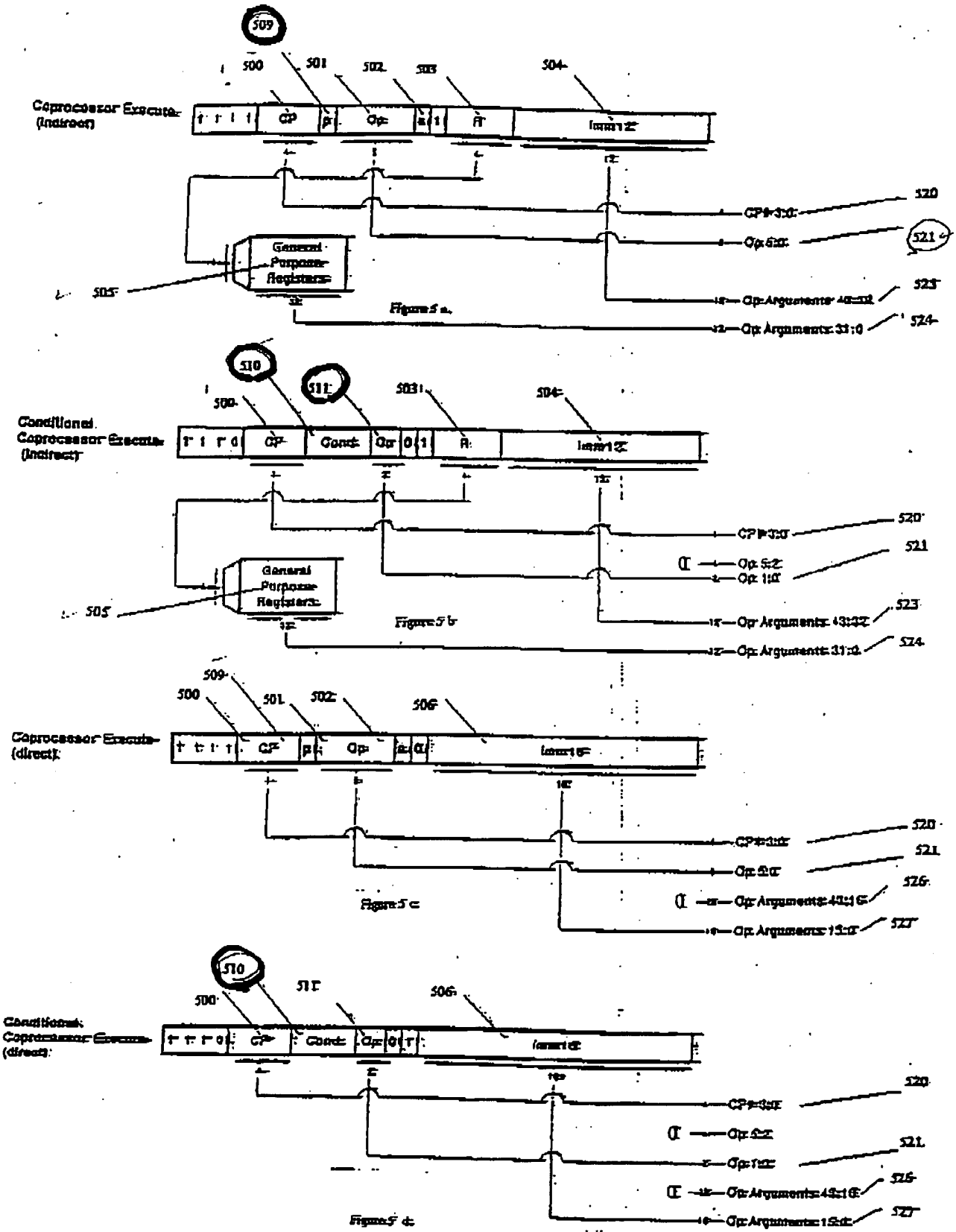

GCH/GDH
CLP
88
Coprocessor #1
96
Scalar Registers
90
Array Registers
102
80
GPRS
Coprocessor #2
98
Scalar Registers
92
GPRS
84
Scalar Registers
Array Registers
104
86
Array Registers
Coprocessor #15
82
100
Scalar Registers
94
Array Registers
106

FIGURE 3


509
500
501
502
503
504
Coprocessor Execute (Indirect)
CP
Op
R
Immediate
520
521
General Purpose Registers
505
Op Argument 63:32
523
Op Argument 31:0
524
Figure 5a.
510
511
503
Conditional Coprocessor Execute (Indirect)
Cond
0
1
Figure 5b
509
500
501
502
506
Coprocessor Execute (direct)
Op
Immediate
526
Op Argument 63:16
527
Figure 5c
510
511
506
Conditional Coprocessor Execute (direct)
Figure 5d.